Title: Test Information Management System. Report, FY 2020 and FY 2021.

Author(s): Benedikt, Bartlomiej Jozef

Intended for: Report

Issued: 2021-09-30

Test Information Management System

Report, FY 2020 and FY 2021

Written by Bart Benedikt, W-13

UNCLASSIFIED

Executive Summary

Delivery Environments supported work in Fiscal Years 2020 and 2021 to develop the Test Information Management System (TIMS). This effort involved creating a suite of tools consisting of a database and the uploading and downloading scripts. It has been demonstrated that the TIMS database is suitable for archiving raw measurements performed on various measured test articles. The TIMS database has been populated with data from the TRUST projects in FY 20 and FY 21. This included the functional data in the Sensors records as well as metadata stored in records in several other tables.

The performed work demonstrated that a database archiving validation test data and integrating it with engineering analysis simulations has a great potential to increase the efficiency, responsiveness, and confidence in the modeling, simulation, and validation efforts for future (and current) weapon systems and assemblies in normal and abnormal environments. In this report, we also discuss how the records belonging to the same project should be arranged, what are the minimum requirements for linking the records using the tabular links, and we provide several recommendations to the projects engineers and the database administrator to improve the TIMS database.

Table of Contents

1 Introduction 7
2 Test Information Management System (TIMS) 8
2.1 TIMS Database Overview 8
2.1.1 Projects Table 9
2.1.2 Testing Series Table 10
2.1.3 Tests Table 11
2.1.4 Sensors Table 12
2.1.5 Data Files Table 12
2.1.6 Parts Table 12
2.1.7 Part Instances Table 13
2.1.8 Documents Table 13
2.2 Granta MI: Scripting Toolkit 2.0 14
2.3 TIMS Example Project: TRUST Thermal Contact Conductance 16
2.3.1 How data is arranged in TIMS database 17
2.3.2 Projects 18
2.3.3 Testing Series 20
2.3.4 Tests 22
2.3.5 Sensors 25
3 Recommendations 28
4 Accomplishments and Future Work 29
5 Acknowledgements 30
6 References 30
7 Appendix A – Python Scripts Used for Searching and Exporting TIMS Data 31
8 Appendix B – List and Description of Attributes 35
8.1 General Record Information 35
8.2 Project Table 36
8.3 Testing Series Table 37
8.4 Tests Table 38
8.5 Sensors Table 39
8.6 Data Files Table 41
8.7 Parts Table 42
8.8 Part Instances Table 43

8.9 Documents Table 44
9 Appendix C – List of Acronyms and Abbreviations 45

List of Figures

Figure 1 Overview of TIMS database tabular structure. There are eight distinct tables in TIMS database. 8
Figure 2 Tabular links between records belonging to different tables..9
Figure 3 Tabular attributes showing links in Test Series record: Related Documents & Images and Link to Test Records..11
Figure 4 Script used to import STK package. These lines should be added to Python script that needs to access TIMS database. 14
Figure 5 Script used to connect to TIMS database that uses Z number and crypto-card password credentials..14
Figure 6 Script used to connect to TIMS database that uses Z number and WIN password credentials...15
Figure 7 Unit systems available in the database can be listed using the unit_systems_available method. 15
Figure 8 Method set_unit_system applied to the database object changes the default unit system. 15
Figure 9 Schematic representation of the experimental set up used to measure the thermal conductivity of the interface between two cylindrical specimens (denoted as specimen A and specimen B) as a function of various magnitudes of axial loading. 16
Figure 10 Schematic view showing how records in the TIMS database are required to be saved. Each Sensor record must have a Test parent, each Test record must have a Test Series parent, and finally each Test Series record must have a Project parent. 17
Figure 11 Tabular attribute 'Link to Testing Series Records' in 'TRUST: Thermal Contact Conductance' project record contains only one link to Test Series table records. 18
Figure 12 View of the window used to edit Link to Testing Series Records attribute 18
Figure 13 Example script used to find the Test Series record using the tabular attribute 'Link to Testing Series Records' from the main project record. Custom functions get_records and get_record_names_from_tabular are explained in the Appendix. 19
Figure 14 Variable test_series_records_list is populated with a single Test Series record object. This is the same record as seen in Figure 13 as expected. 19
Figure 15 Tabular attribute 'Link to Test Records' in 'MST-8 Right Cylinder' test series record contains links to individual test records. 20
Figure 16 Example script used to find the Test records using the tabular attribute 'Link to Test Records' from the test series record. 21
Figure 17 Variable test_records_list is populated with individual Test record objects (same as records listed in **Figure 15**). 21
Figure 18 Tabular attribute 'Sensors Used in this Test' in SS-AL-20191211_1p6_15N test record contains links to base and derived sensor records...23
Figure 19 Example script used to find the Sensor records using the tabular attribute 'Sensors Used in this Test' from the test record. 24
Figure 20 Variable sensor_records_list is populated with individual sensor record objects (same as records listed in Figure 18)..24
Figure 21 Typical information included in the sensor record window. Data for sensor TC_6_SS-AL-20191211_1p6_15N is shown...25
Figure 22 Temperature vs. time plot for sensor TC_6_SS-AL-20191211_1p6_15N 26

Figure 23 Script to read and plot temperature vs. time data for the first six thermocouples from Test SS-AL-20191211_1p6_15N 27
Figure 24 Plot generated by script shown in **Figure 23**. Temperature vs. time curves for six sensors from Test SS-AL-20191211_1p6_15N is shown. 27
Figure 25 Script used to start STK 2.x package and to connect to TIMS database 31
Figure 26 Example script to define the custom functions get_records and get_record_names_from_tabular to search the TIMS database using the tabular links. 32
Figure 27 Example script to define the custom function get_functional_data_dict to import sensor's functional data to Python dictionary. 33
Figure 28 Script to search: Project, Test Series, Tests, and Sensor records using the tabular links. 34

List of Tables

Table 1. General Record Information attributes used in any record. 35
Table 2. Attributes used in Project records 36
Table 3 Attributes used in Testing Series records 37
Table 4 Attributes used in Tests records 38
Table 5 Attributes used in Sensors records (1) 39
Table 6 Attributes used in Sensors records (2) 40
Table 7 Attributes used in Data Files records 41
Table 8 Attributes used in Parts records 42
Table 9 Attributes used in Part Instances records 43
Table 10 Attributes used in Documents records 44

1 Introduction

Test Information Management System (TIMS) is a suite of tools consisting of a database and the uploading and downloading scripts. The test data of interest in this project could be various measurements (e.g. displacements, forces, temperature, etc.) taken at discrete points on a thermally and/or mechanically loaded engineering part or assemblies. More specifically, the database is designed to provide the data used for the Validation Suite (VS) collection of models that support validation work for engineering analysis simulations in W-13 group.

In contrast to the Material Information Management System (MIMS) Granta database for materials widely used at Los Alamos National Laboratory, the TIMS database is used to save raw measurements as a function of time, information regarding the exact location and orientations of the measurement devices (a.k.a. sensors), and information on the part geometry and loading conditions. Unlike MIMS, the tests in TIMS are not intended specifically to calibrate material model parameters or advance constitutive model development. Instead, the data in TIMS is used to provide evidence for the validation of engineering analysis simulations and to provide inputs for validation procedures that may result in improvement in the estimates of model input parameters. This process involves transferring part and loading information into a Finite Element (FE) code, numerically simulating the test, and comparing the simulated results with the measured data at the testing locations (where the sensors were attached to a part). The main purpose of the work presented in this report was to set up a database suitable for storing raw and 'cleaned' measurements performed at discrete points and to demonstrate that this database can be coupled with Finite Element analysis software.

All numerical examples considered in this proof-of-concept stage of the project were taken from the TRUST: Thermal Contact Conductance project. The TRUST: Thermal Contact Conductance project was chosen because it is conceptually simple and the measurements results have been saved in TIMS recently.

The presented project leverages work already underway on the MIMS, which makes material tests, properties, models, and related information available to engineers and scientists across LANL. The TIMS work falls outside the scope of MIMS because it does not involve coupon-level material tests. As pointed out above TIMS is intended to capture test data and the test related metadata (e.g. part information, location of sensors, etc.) used for validating and calibrating engineering weapon models. It also does not fall under the purview of the Echo data analysis software, which is currently used specifically for environmental specification analyses. Making such test data available is critical to ensuring the quality and responsiveness of weapons engineering calculations such as those used to quantify the effects of specific thermal and mechanical environments on the response of weapon components.

2 Test Information Management System (TIMS)

2.1 TIMS Database Overview

TIMS database is divided into eight tables, as shown in Figure 1. A table is a collection of records that can all be described using the same attributes (all attributes for each table are listed in Appendix B – List and Description of Attributes). For example, records in the Sensors table contain attributes describing the sensor location and the data collected by the sensor, while records in the Projects table contain attributes describing the project name, the principal investigators and the funding source. Arranging the data this way accomplishes two goals: first, it minimizes data duplication and second, it allows to find the required data quickly. For example, if stored in the same record as sensor data, the project name would be duplicated many times. Instead the project name is stored in the Project record (i.e. in a record in the Projects table) and this Project record is linked to all records in other tables that are associated with that project, including the Sensor record, which contains the sensor data. Inside each table, a sub-directory structure can be created to arrange the records in a logical way.

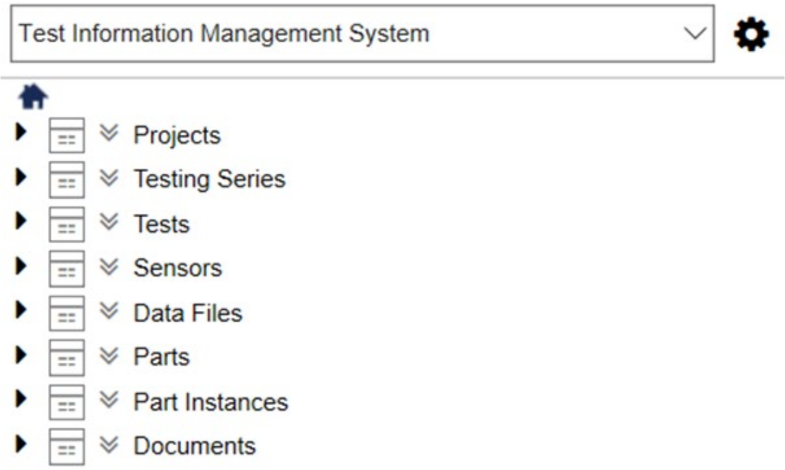


Figure 1 *Overview of TIMS database tabular structure. There are eight distinct tables in TIMS database.*

Links are used to ensure the user has access to all information related to test data, no matter in which table the information is stored. There are two types of links: *static links* are created manually and do not change based on the data stored in record attributes, while *linked tabular data attributes* are links defined by the records unique IDs. Linked tabular data attributes rely on the 'ID' attributes, called *tabular linking values* that are unique to a given record. They are used to tell a linked tabular data attribute which records to link to. For example a Testing Series record contains an attribute called 'Testing Series ID'. Linking to a Testing Series record from, say, a Project record is achieved by suppling the linked tabular data attribute in the Project record with the value of a 'Testing Series ID' in a Testing Series record. The linked tabular data attribute in the Project record will then search the database for Testing Series records containing that specific value of 'Testing Series ID' and create a link to it. In this way, a single linked tabular data attribute may link to multiple records from a single table.

A schematic view of the available tabular data links between records in different tables is presented in Figure 2. There is a somewhat a hierarchical nature to the structure. A Project record is the highest level and is the parent of at least one Testing Series record, which is in turn the parent of at least one (and often many) Tests records. Tests records contain test metadata and the Test records are the parents of at least one (and often many) Sensor records, which contain the actual measured data. The Data Files from which the Sensor data and Tests data originate are stored in their own table and they are linked to the Sensors records. Since it is important to distinguish which Part (i.e. 'part number') and which Part Instance (i.e. 'serial number') a Sensor is attached to, this information is stored in separate tables and linked to Testing Series and Projects. Finally, there may be various documents that provide supporting information to the test data, and these are stored in the Documents table and linked to the Projects, Testing Series, and Tests records.

It should be noted that the TIMS database is intended to store and relate test data for easy retrieval, both by automated scripting (e.g. for model developers using Python) and by use of the relatively intuitive GUI (e.g. for occasional users). In particular, the TIMS database is *not* intended as an analysis tool for the reduction of data from raw 'uncleaned' data to useful 'cleaned' data. However, it is intended that the processes by which the 'cleaned' data in TIMS were produced be documented in TIMS.

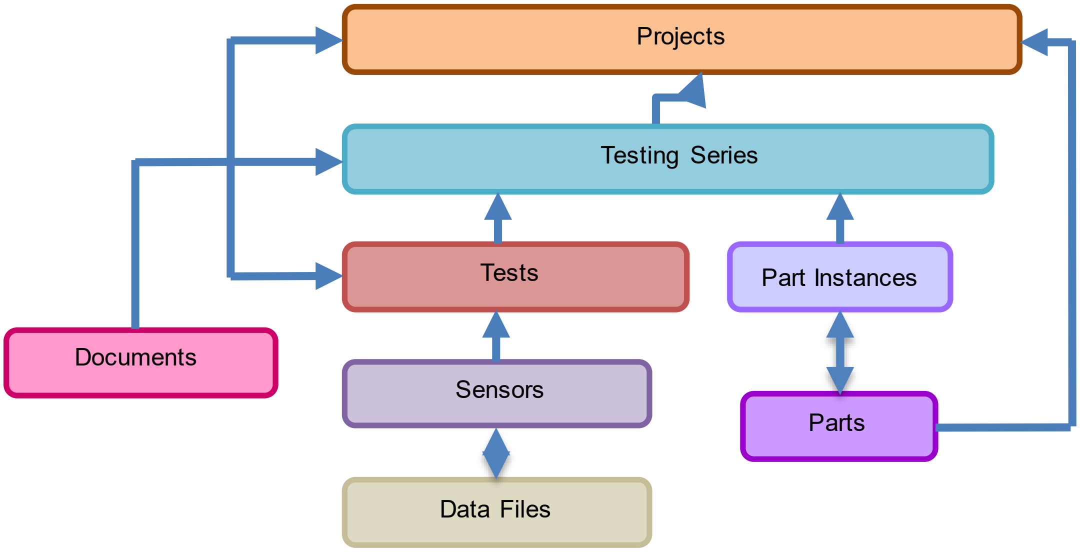


Figure 2 *Tabular links between records belonging to different tables.*

2.1.1 Projects Table

At the highest level of the database structure is the Project table, which is intended to describe a set of tests, usually supported by a single program, to achieve a well-defined objective. A Project contains at least one Testing Series, although multiple Testing Series may be necessary.

The records in the Project table use tabular linking attributes to group records associated with a single project (e.g. test data, parts information, documentation etc.). The Project record has the following tabular attributes to define links to other records associated with the project:

- **Link to Parts**: links to all parts tested in the project

- **Link to Testing Series Record**: links to all test series performed on any part that belongs to the project
- **Link to Documentation**: links to all documents supporting or describing parts or tests performed in the projects

A list of all attributes used in Project records is given in Table 2.

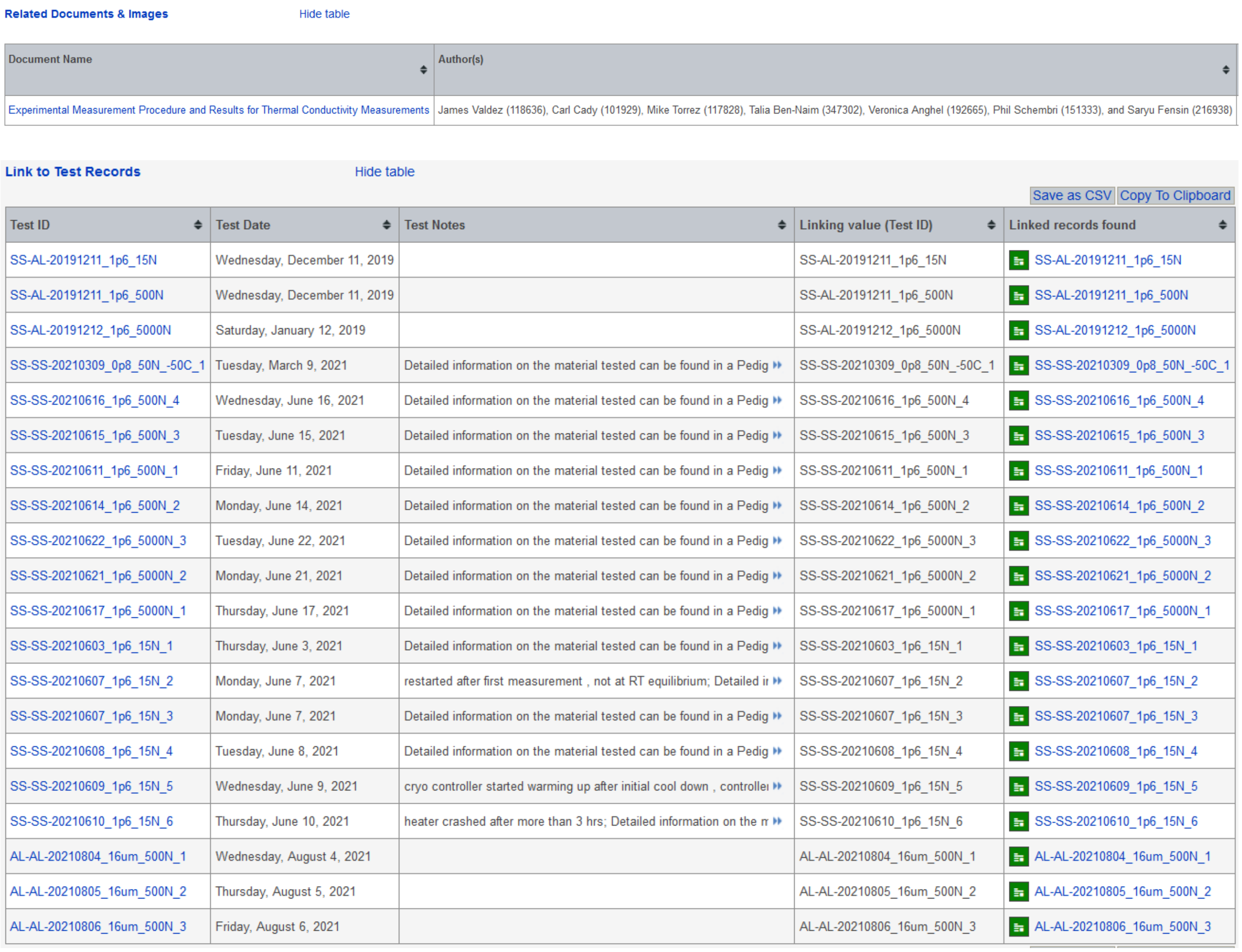
2.1.2 Testing Series Table

It is often useful to separate individual tests in a Project into smaller groups known as Testing Series based on the similarities between the tests. For example, if a project consisted of 20 tests; 12 mechanical tests and 8 thermal tests, then this difference forms a natural way to define two distinct Testing Series within the project: one for mechanical tests and one for thermal tests. Additionally, it may be desirable to collect a group of tests into a Testing Series in which a specific set of factors, e.g. test temperature or loading magnitude, were varied. Gathering such tests into a single Testing Series provides a convenient way for users to find, retrieve, and compare the results of these related tests.

The following linked tabular attributes are used in Testing Series records to link records saved in other tables with the Testing Series record:

- **Related Documents & Images:** links to Documents records
- **Link to Test Records**: links to individual Test records
- **Link to Part Instances**: links to individual Part Instances

Examples of tabular link attributes in a record belonging to the Testing Series are shown in Figure 3.



Related Documents & Images Hide table

Document Name	Author(s)
Experimental Measurement Procedure and Results for Thermal Conductivity Measurements	James Valdez (118636), Carl Cady (101929), Mike Torrez (117828), Talia Ben-Naim (347302), Veronica Anghel (192665), Phil Schembri (151333), and Saryu Fensin (216938)

Link to Test Records Hide table

Save as CSV | Copy To Clipboard

Test ID	Test Date	Test Notes	Linking value (Test ID)	Linked records found
SS-AL-20191211_1p6_15N	Wednesday, December 11, 2019		SS-AL-20191211_1p6_15N	SS-AL-20191211_1p6_15N
SS-AL-20191211_1p6_500N	Wednesday, December 11, 2019		SS-AL-20191211_1p6_500N	SS-AL-20191211_1p6_500N
SS-AL-20191212_1p6_5000N	Saturday, January 12, 2019		SS-AL-20191212_1p6_5000N	SS-AL-20191212_1p6_5000N
SS-SS-20210309_0p8_50N_-50C_1	Tuesday, March 9, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210309_0p8_50N_-50C_1	SS-SS-20210309_0p8_50N_-50C_1
SS-SS-20210616_1p6_500N_4	Wednesday, June 16, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210616_1p6_500N_4	SS-SS-20210616_1p6_500N_4
SS-SS-20210615_1p6_500N_3	Tuesday, June 15, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210615_1p6_500N_3	SS-SS-20210615_1p6_500N_3
SS-SS-20210611_1p6_500N_1	Friday, June 11, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210611_1p6_500N_1	SS-SS-20210611_1p6_500N_1
SS-SS-20210614_1p6_500N_2	Monday, June 14, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210614_1p6_500N_2	SS-SS-20210614_1p6_500N_2
SS-SS-20210622_1p6_5000N_3	Tuesday, June 22, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210622_1p6_5000N_3	SS-SS-20210622_1p6_5000N_3
SS-SS-20210621_1p6_5000N_2	Monday, June 21, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210621_1p6_5000N_2	SS-SS-20210621_1p6_5000N_2
SS-SS-20210617_1p6_5000N_1	Thursday, June 17, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210617_1p6_5000N_1	SS-SS-20210617_1p6_5000N_1
SS-SS-20210603_1p6_15N_1	Thursday, June 3, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210603_1p6_15N_1	SS-SS-20210603_1p6_15N_1
SS-SS-20210607_1p6_15N_2	Monday, June 7, 2021	restarted after first measurement , not at RT equilibrium; Detailed in ⏩	SS-SS-20210607_1p6_15N_2	SS-SS-20210607_1p6_15N_2
SS-SS-20210607_1p6_15N_3	Monday, June 7, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210607_1p6_15N_3	SS-SS-20210607_1p6_15N_3
SS-SS-20210608_1p6_15N_4	Tuesday, June 8, 2021	Detailed information on the material tested can be found in a Pedig ⏩	SS-SS-20210608_1p6_15N_4	SS-SS-20210608_1p6_15N_4
SS-SS-20210609_1p6_15N_5	Wednesday, June 9, 2021	cryo controller started warming up after initial cool down , controlle ⏩	SS-SS-20210609_1p6_15N_5	SS-SS-20210609_1p6_15N_5
SS-SS-20210610_1p6_15N_6	Thursday, June 10, 2021	heater crashed after more than 3 hrs; Detailed information on the m ⏩	SS-SS-20210610_1p6_15N_6	SS-SS-20210610_1p6_15N_6
AL-AL-20210804_16um_500N_1	Wednesday, August 4, 2021		AL-AL-20210804_16um_500N_1	AL-AL-20210804_16um_500N_1
AL-AL-20210805_16um_500N_2	Thursday, August 5, 2021		AL-AL-20210805_16um_500N_2	AL-AL-20210805_16um_500N_2
AL-AL-20210806_16um_500N_3	Friday, August 6, 2021		AL-AL-20210806_16um_500N_3	AL-AL-20210806_16um_500N_3

Figure 3 *Tabular attributes showing links in Test Series record: Related Documents & Images and Link to Test Records*

A list of all attributes used in Testing Series records is given in Table 3.

2.1.3 Tests Table

Test metadata, such as the test date and ambient temperature, is stored in records in the Tests table. A test is defined as a collection of sensor results obtained from the same part/assembly at the same time. Note that the test data itself is stored in the Sensors records, and so the Tests record serves to gather together, using the linked tabular data attribute 'Link to Sensor Records', all Sensors records corresponding to a single test.

The following linked tabular attributes are used in Test records to link records saved in other tables with the Test record:

- **Documents:** links to Documents records
- **Sensors Used in this Test**: links to individual Sensor records

A list of all attributes used in Tests records is given in Table 4.

2.1.4 Sensors Table

The individual measurements at a point as a function of time are stored in the Sensors table records. A Sensors record corresponds to a single sensor output, which is assumed to be a function of time, and the data is stored in *functional data* attributes, which are structured as x-y pairs.

A Sensors record should be used to contain only one set of functional data. For example, if a given test consists of measuring force in two locations, displacement in one location, and temperature in three locations, then the measured data will be saved in 6 separate Sensors records (2 + 1 + 3 = 6). All of these Sensors records will be linked to the same Tests record.

A virtual sensor record can also be created in the Sensors table. The purpose of the virtual sensor is, for example, to down-sample the number of data points or to algebraically combine information from various sensors. It is also allowed that the data saved in a virtual sensor is computed using a numerical model only. A virtual sensor is labeled as Derived in the 'Base or Derived' attribute while a physical sensor is labeled as Base. Note that Granta does not contain functions for processing Sensors data to create virtual Sensors data; such processing is performed externally and then documented in Granta (for example in the Documents table) for traceability.

In a Sensors record, the following linked tabular attribute 'Link to Data File Records' is used to link the data in the record to the data file from which it originates:

- **Link to Data Files**: links to Data File record

A list of all attributes used in Sensors records is given in Table 5.

2.1.5 Data Files Table

Data File records are used to store the raw data files from which the data in Sensors records originates. Note that the data for multiple Sensors records may have originated from the same data file, and so one Data Files record may be associated with multiple Sensors records. In addition to the attached files, the Data Files records may also provide information regarding how the files are formatted (e.g. delimiter, header information, etc.) or they may provide physical time when the measurements were started.

As a general rule, the numerical information in the raw data files and the functional information in Sensors records should be the same. Changes to the data such as filtering or time-shifting should be achieved by creating virtual (a.k.a. Derived) Sensors records and documenting the processes used to create the data from the physical (a.k.a. Base) Sensors records.

The files stored in Data Files records are intended to document the source of the data for traceability. It is intended that data be retrieved from the Sensors records and not from the files in the Data Files record.

In a Data Files table record, the 'Link to Sensor Records' linked tabular attribute links to individual sensors associated with the data files.

A list of all attributes used in Data Files records is given in Table 7.

2.1.6 Parts Table

Information, such as part numbers, drawing numbers, and a link to PDMLink, about parts on which Sensors are attached is stored in Parts records. Part information is stored in a dedicated table rather than in, for example, the Testing Series (or Projects) table because there may be multiple Testing Series (or Projects) performed using the same part, and storing the part information in its own record avoids duplication of the information in multiple Testing Series (or Projects) records.

A list of all attributes used in Parts records is given in Table 8.

2.1.7 Part Instances Table

Information about individual instances of a part is stored in Part Instances records. Currently this information consists of the serial number of the part but could be expanded to include information about, for example, environmental exposures of particular part instances. As with the Parts table, this information is broken out into its own table to avoid duplication in other database records.

A list of all attributes used in Part Instances records is given in Table 9.

2.1.8 Documents Table

Various documents describing the parts, testing environment, modeling, etc. can be stored in the Documents table. Such documents provide traceability and supply important metadata that may not be stored in in other attributes. For example, a test report is a document that provides information about testing conditions that may not be completely captured in the attributes in the Tests record, and these conditions may be important when comparing the test results to simulation results.

A list of all attributes used in Documents records is given in Table 10.

2.2 Granta MI: Scripting Toolkit 2.0

In this section, a brief description of the Granta Scripting Toolkit (STK) for Python is provided. STK 2.x is used to provide the interface between Python scripting language and Granta databases. In this work, we will focus on the interface between Python and TIMS database. At present, the STK package can be run from Python 3 only. Older versions of Python (e.g. 2.x) are not supported.

For the LANL users, the complete STK documentation can be found in the following document: https://int.lanl.gov/projects/W13docs/GrantaSTK. In the next paragraph, the steps needed to import the STK module and connect to the TIMS database from the Python script are presented.

LANL users who have access to W-13 Python Environment and can run local Linux machine or access sstelmo server can access the STK module by following these steps:

- Load the anaconda python environment:

```
module load python/2020.07-python-3.8
```

- Source the anaconda environment:

```
sv3r
```

Once the W-13 Python Environment is loaded and activated, a LANL user will have access to STK package.

If the W-13 Python Environment is not available then the STK package can be installed following the steps described in the User Manual.

To access/load the STK package from a Python script, the package has to be imported by adding two lines shown in **Figure 4** to the beginning of the script. It is strongly recommended that the short hand notation used in **Figure 4** is followed in any script that uses STK interface.

```
import GRANTA_MIScriptingToolkit as GrantaToolkit
from GRANTA_MIScriptingToolkit import granta as mpy
```

Figure 4 *Script used to import STK package. These lines should be added to Python script that needs to access TIMS database.*

The next step is to open the connection between Python and the TIMS database. To accomplish this, the user has to provide the proper credentials. At present, two authentication methods are allowed.

In the first method, a LANL user must provide the Z number and the password generated by the crypto-card. An example script used to connect to TIMS database that uses this method is given in **Figure 5**.

```
import getpass
your_Z_number  = str(input('Enter Your Z number: '))
password       = getpass.getpass('Enter your crypto password: ')
mi = mpy.connect("http://grantami/mi_servicelayer", user_name=your_Z_number,\
        password=password,  domain="lanl.gov", autologon=False)
my_db = mi.get_db("Test Information Management System")
```

Figure 5 *Script used to connect to TIMS database that uses Z number and crypto-card password credentials.*

In the second method, a user is authenticated using the Z number and a temporary WIN password generated from https://register.lanl.gov. An example script used to connect to TIMS database that used this method is shown in **Figure 6**.

```
import getpass
your_Z_number  = str(input('Enter Your Z number: '))
password = getpass.getpass('Enter your WIN temporary password: ')
mi = mpy.connect("http://grantami/mi_servicelayer",user_name=your_Z_number, \
        password=password, domain="win.lanl.gov", autologon=False)
my_db = mi.get_db("Test Information Management System")
```

Figure 6 *Script used to connect to TIMS database that uses Z number and WIN password credentials.*

Values of the measured physical quantities stored in the TIMS database can be listed using various unit systems. Available unit systems defined for the accessed database can be displayed using the *unit_systems_available* method. Applying this method to the database object returns a list of names of the unit systems defined in the given database (see **Figure 7**).

```
In [1]: my_db.unit_systems_available
Out[1]:
['CGS (Consistent)',
 'FPS (Consistent)',
 'IPS (Consistent)',
 'Metric',
 'mmNs (Consistent)',
 'SI (Consistent)',
 'UK Imperial',
 'US Customary']
```

Figure 7 *Unit systems available in the database can be listed using the unit_systems_available method.*

The requested unit system for a database object can be specified using the *set_unit_system* method as shown in **Figure 8**. It is recommended that the unit system is always specified before any data from the database object is downloaded to Python. Note that the parameter absolute_temperature, if set to True, makes the requested output use absolute temperature units.

```
my_db.set_unit_system('SI (Consistent)', absolute_temperatures=True)
```

Figure 8 *Method set_unit_system applied to the database object changes the default unit system.*

2.3 TIMS Example Project: TRUST Thermal Contact Conductance

The *TRUST: Thermal Contact Conductance* project was chosen to illustrate how STK interface can be used to access TIMS data. The testing performed in the project are described in detail in (Brindley, LeBrun, & Lum, 2020). In this work only a brief summary of testing work is provided.

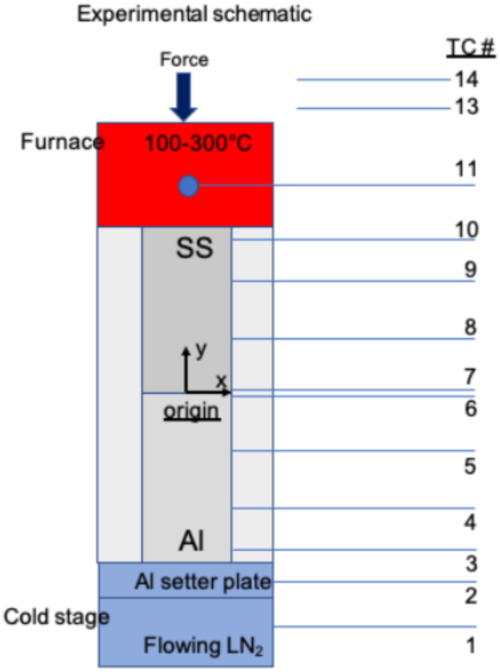


Figure 9 Schematic representation of the experimental set up used to measure the thermal conductivity of the interface between two cylindrical specimens (denoted as specimen A and specimen B) as a function of various magnitudes of axial loading.

The main purpose of the experimental work saved in *TRUST: Thermal Contact Conductance* project was to measure the thermal conductivity of the interface between two cylindrical specimens. The specimens were made from metal such as aluminum or steel. They were machined as long solid cylinders with equal diameter. The cylinders were placed between the loading platens on top of each other. Each platen was capable of maintaining various temperatures. Also, the platens were used to compress the cylindrical specimens axially and the axial compressive load could be controlled and monitored. Finally, a number of thermocouples (TC) were bonded to each cylindrical specimen, so that the temperature distribution along the axial direction of each specimen could be monitored. The schematic representation of the described tests is shown in **Figure 9**. Based on the measured temperature distribution in both specimens and based on the known temperature of the loading platens, the thermal properties of the interface between the specimens could be determined. By repeating the described experiments using different values of axial loading and different specimens (e.g. made from different material or having different surface roughness) the interface thermal conductivity can be measured for various conditions. More details about the thermal conductivity experiments can be also found in (Valdez, et al., 2020).

2.3.1 How data is arranged in TIMS database

It is recommended that any test data saved in the TIMS database should be arranged in such a way to follow the schematic shown in **Figure 10**.

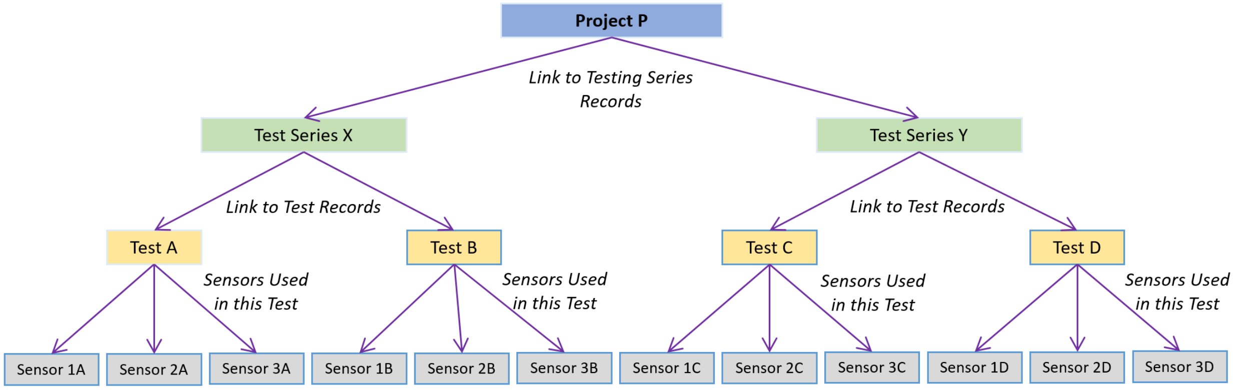


***Figure 10** Schematic view showing how records in the TIMS database are required to be saved. Each Sensor record must have a Test parent, each Test record must have a Test Series parent, and finally each Test Series record must have a Project parent.*

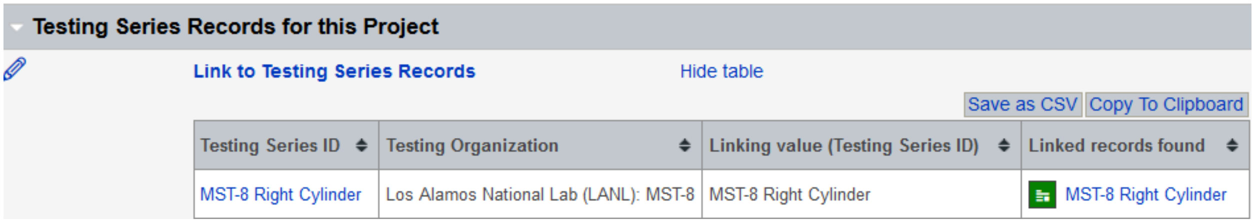
At the top level, there are projects. Each project record should have a tabular attribute *Link to Testing Series Records* populated with the tabular links to its children Testing Series records. Next, each Testing Series record should have a tabular attribute *Link to Test Records* populated with links to its children Test records. Finally, each Test record should have tabular attribute *Sensors Used in this Test* populated with links to individual Sensor records. It is very important to follow this arrangement because it allows to quickly search the database and to understand relationships between various Sensor and Test records. Even if a Project, or Test Series, or Test record has only a single child, the arrangement of records shown in **Figure 10** should still be followed.

In sections 2.3.2 to 2.3.5 it is shown and discussed how the measured data in various tables and records can be accessed using the STK interface and how the tabular links can be created to follow the data arrangement shown in **Figure 10**. As already mentioned, the Thermal Contact Conductance project is used as an example.

2.3.2 Projects

The main purpose of the project record is to gather in one record the links to all test series records, and if applicable to documents and parts records. In addition, the general information about the tests belonging to a given project should also be provided in the project level record. It is mostly up to the responsible engineer to decide how various tests can be divided into projects and what specific information about the tests is relevant enough to be uploaded to the TIMS database.

It is recommended that any released project record has tabular attribute *Link to Testing Series Records* populated with the links to its children Testing Series records. This tabular attribute for *TRUST: Thermal Contact Conductance* project record is shown in **Figure 11**. It is seen that the project record has only one child Testing Series record.



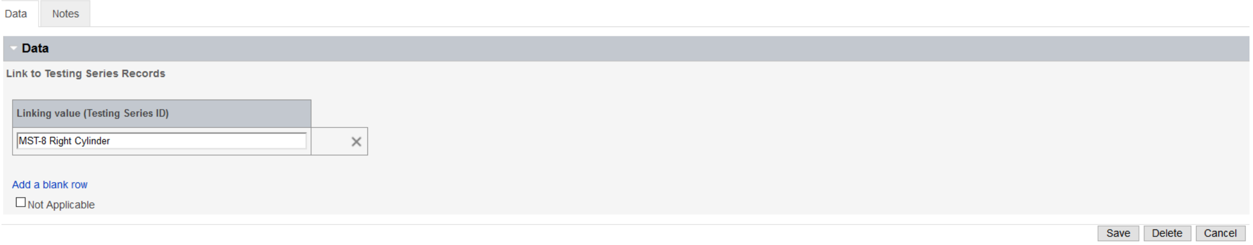
Testing Series Records for this Project

Link to Testing Series Records

Testing Series ID	Testing Organization	Linking value (Testing Series ID)	Linked records found
MST-8 Right Cylinder	Los Alamos National Lab (LANL): MST-8	MST-8 Right Cylinder	MST-8 Right Cylinder

Figure 11 *Tabular attribute 'Link to Testing Series Records' in 'TRUST: Thermal Contact Conductance' project record contains only one link to Test Series table records.*

To populate and/or edit tabular attribute *Link to Testing Series Records* one has to click on the pencil icon to the left from the name of the attribute. Once this icon is clicked, the user will be able to edit the window shown in **Figure 12**.



Data

Link to Testing Series Records

Linking value (Testing Series ID)
MST-8 Right Cylinder

☐ Not Applicable

Figure 12 *View of the window used to edit Link to Testing Series Records attribute*

To link Testing Series records and include them in the *Link to Testing Series Records* table, each *Testing Series ID* has to be entered to the *Linking value (Testing series ID)* field as seen in **Figure 12**. If additional records have to be linked, the *Add a blank row* is used to create additional rows to link additional Test Series records. After all records are added, the save button is used to save the changes.

It is important to remember that in order to use this method each linked Test Series record must have the valid attribute *Testing Series ID* filled out. The *Testing Series ID* attribute must be unique for each record to avoid confusion and linking wrong records to the project.

Finally, we show how to use the STK interface to access the project record and find test series records linked to the project. After following the steps described in section 2.2 to log in to the TIMS database, the user needs to open the Project table object, find the Thermal Contact Conductance project record, and read the linked records from the tabular attribute *Link to Testing Series Records*. The script to perform these steps is shown in **Figure 13**.

The script shown in **Figure 13** uses two custom functions, namely *get_records* and *get_record_names_from_tabular*. These functions use STK functionality and they are explained in the Appendix. In short, function *get_records* returns a list of record objects that have the name attribute equal to either element of *rec_name_project* input list and are located in the table object *tbl_project*. Function *get_record_names_from_tabular* returns a list of record objects from the tabular attribute defined by *table_attr_name_in_project* and belonging to record object *project_records_list*[0].

```
tbl_project = my_db.get_table("Projects")
#
# Name of the project record:
# --------------------------
rec_name_project = ['TRUST: Thermal Contact Conductance']
#
# List of Project records found:
# -----------------------------
project_records_list = get_records(rec_name_project, tbl_project)
#
# Name of the tabular attribute with all linked Test Series records:
# ----------------------------------------------------------------
table_attr_name_in_project = 'Link to Testing Series Records'
#
# Find Test Series records linked from the Project record:
# ------------------------------------------------------
test_series_records_list =
get_record_names_from_tabular([project_records_list[0]],\
      table_attr_name_in_project)
```

Figure 13 *Example script used to find the Test Series record using the tabular attribute 'Link to Testing Series Records' from the main project record. Custom functions get_records and get_record_names_from_tabular are explained in the Appendix.*

After the script shown in **Figure 13** is executed, variable *test_series_records_list* is populated with the record object shown in **Figure 14**. The content of this list is the same as the content of *Link to Testing Series Records* table shown in **Figure 11**, as expected.

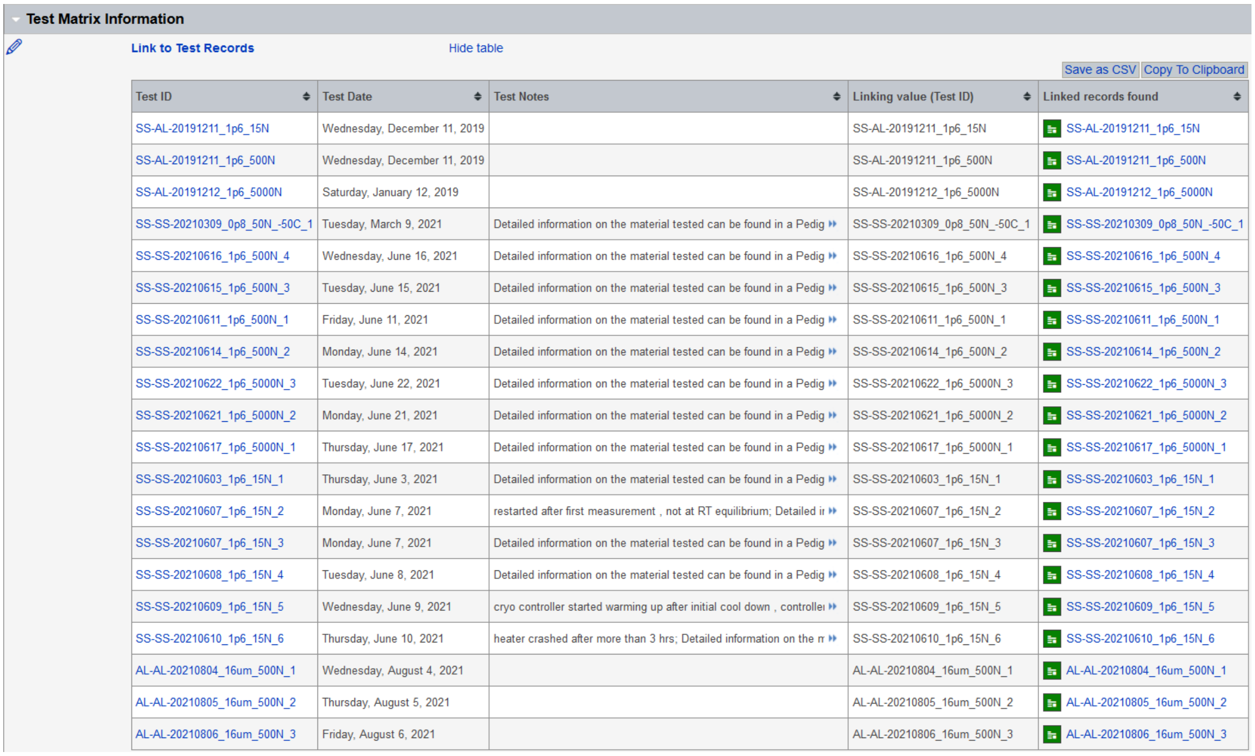
```
In [3]: test_series_records_list
Out[3]: [<Record long name:MST-8 Right Cylinder>]
```

Figure 14 *Variable test_series_records_list is populated with a single Test Series record object. This is the same record as seen in Figure 13 as expected.*

2.3.3 Testing Series

Test series records are located one level below the project record. Each test series record must be linked with the project record via *Testing Series ID* attribute as explained in the previous section. Also, each test series record must be linked to its children test record(s). The purpose of the test series record is to provide a tool to group similar tests together. It is up to the project engineer to provide rationale for grouping the individual tests into test series records.

Following the *Thermal Contact Conductance* project we open the test series record named *MST-8 Right Cylinder*. As explained before, it is the only test series record for this project. The dynamic links to the individual tests are given in tabular attribute *Link to Test Records*. The table showing these individual records is shown in **Figure 15**.



Test ID	Test Date	Test Notes	Linking value (Test ID)	Linked records found
SS-AL-20191211_1p6_15N	Wednesday, December 11, 2019		SS-AL-20191211_1p6_15N	SS-AL-20191211_1p6_15N
SS-AL-20191211_1p6_500N	Wednesday, December 11, 2019		SS-AL-20191211_1p6_500N	SS-AL-20191211_1p6_500N
SS-AL-20191212_1p6_5000N	Saturday, January 12, 2019		SS-AL-20191212_1p6_5000N	SS-AL-20191212_1p6_5000N
SS-SS-20210309_0p8_50N_-50C_1	Tuesday, March 9, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210309_0p8_50N_-50C_1	SS-SS-20210309_0p8_50N_-50C_1
SS-SS-20210616_1p6_500N_4	Wednesday, June 16, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210616_1p6_500N_4	SS-SS-20210616_1p6_500N_4
SS-SS-20210615_1p6_500N_3	Tuesday, June 15, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210615_1p6_500N_3	SS-SS-20210615_1p6_500N_3
SS-SS-20210611_1p6_500N_1	Friday, June 11, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210611_1p6_500N_1	SS-SS-20210611_1p6_500N_1
SS-SS-20210614_1p6_500N_2	Monday, June 14, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210614_1p6_500N_2	SS-SS-20210614_1p6_500N_2
SS-SS-20210622_1p6_5000N_3	Tuesday, June 22, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210622_1p6_5000N_3	SS-SS-20210622_1p6_5000N_3
SS-SS-20210621_1p6_5000N_2	Monday, June 21, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210621_1p6_5000N_2	SS-SS-20210621_1p6_5000N_2
SS-SS-20210617_1p6_5000N_1	Thursday, June 17, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210617_1p6_5000N_1	SS-SS-20210617_1p6_5000N_1
SS-SS-20210603_1p6_15N_1	Thursday, June 3, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210603_1p6_15N_1	SS-SS-20210603_1p6_15N_1
SS-SS-20210607_1p6_15N_2	Monday, June 7, 2021	restarted after first measurement , not at RT equilibrium; Detailed ir »	SS-SS-20210607_1p6_15N_2	SS-SS-20210607_1p6_15N_2
SS-SS-20210607_1p6_15N_3	Monday, June 7, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210607_1p6_15N_3	SS-SS-20210607_1p6_15N_3
SS-SS-20210608_1p6_15N_4	Tuesday, June 8, 2021	Detailed information on the material tested can be found in a Pedig »	SS-SS-20210608_1p6_15N_4	SS-SS-20210608_1p6_15N_4
SS-SS-20210609_1p6_15N_5	Wednesday, June 9, 2021	cryo controller started warming up after initial cool down , controller »	SS-SS-20210609_1p6_15N_5	SS-SS-20210609_1p6_15N_5
SS-SS-20210610_1p6_15N_6	Thursday, June 10, 2021	heater crashed after more than 3 hrs; Detailed information on the m »	SS-SS-20210610_1p6_15N_6	SS-SS-20210610_1p6_15N_6
AL-AL-20210804_16um_500N_1	Wednesday, August 4, 2021		AL-AL-20210804_16um_500N_1	AL-AL-20210804_16um_500N_1
AL-AL-20210805_16um_500N_2	Thursday, August 5, 2021		AL-AL-20210805_16um_500N_2	AL-AL-20210805_16um_500N_2
AL-AL-20210806_16um_500N_3	Friday, August 6, 2021		AL-AL-20210806_16um_500N_3	AL-AL-20210806_16um_500N_3

Figure 15 *Tabular attribute 'Link to Test Records' in 'MST-8 Right Cylinder' test series record contains links to individual test records.*

Test records can be linked to the above tabular attribute in the same way as described in the previous section. The only difference is that this time the linking attribute is *Test ID* attribute in test records.

Again, the individual test record objects can be accessed from Python using the STK interface. The script that does it is shown in **Figure 16**.

```
tbl_test_series = my_db.get_table("Testing Series")
#
# Name of the Test Series record:
# ------------------------------
rec_name_test_series = test_series_records_list[0]
#
test_series_records_list = get_records(rec_name_test_series, tbl_test_series)
#
table_attr_name_in_test_series = 'Link to Test Records'
test_records_list = get_record_names_from_tabular(test_series_records_list, \
                                table_attr_name_in_test_series)
```

Figure 16 *Example script used to find the Test records using the tabular attribute 'Link to Test Records' from the test series record.*

After executing the script shown in **Figure 16** it can be verified that *test_records_list* variable is populated with the same record objects (tests) as shown in **Figure 15**.

```
In [4]: test_records_list
Out[4]:
[['SS-AL-20191211_1p6_15N'],
 ['SS-AL-20191211_1p6_500N'],
 ['SS-AL-20191212_1p6_5000N'],
 ['SS-SS-20210309_0p8_50N_-50C_1'],
 ['SS-SS-20210616_1p6_500N_4'],
 ['SS-SS-20210615_1p6_500N_3'],
 ['SS-SS-20210611_1p6_500N_1'],
 ['SS-SS-20210614_1p6_500N_2'],
 ['SS-SS-20210622_1p6_5000N_3'],
 ['SS-SS-20210621_1p6_5000N_2'],
 ['SS-SS-20210617_1p6_5000N_1'],
 ['SS-SS-20210603_1p6_15N_1'],
 ['SS-SS-20210607_1p6_15N_2'],
 ['SS-SS-20210607_1p6_15N_3'],
 ['SS-SS-20210608_1p6_15N_4'],
 ['SS-SS-20210609_1p6_15N_5'],
 ['SS-SS-20210610_1p6_15N_6'],
 ['AL-AL-20210804_16um_500N_1'],
 ['AL-AL-20210805_16um_500N_2'],
 ['AL-AL-20210806_16um_500N_3']]
```

Figure 17 *Variable test_records_list is populated with individual Test record objects (same as records listed in **Figure 15**).*

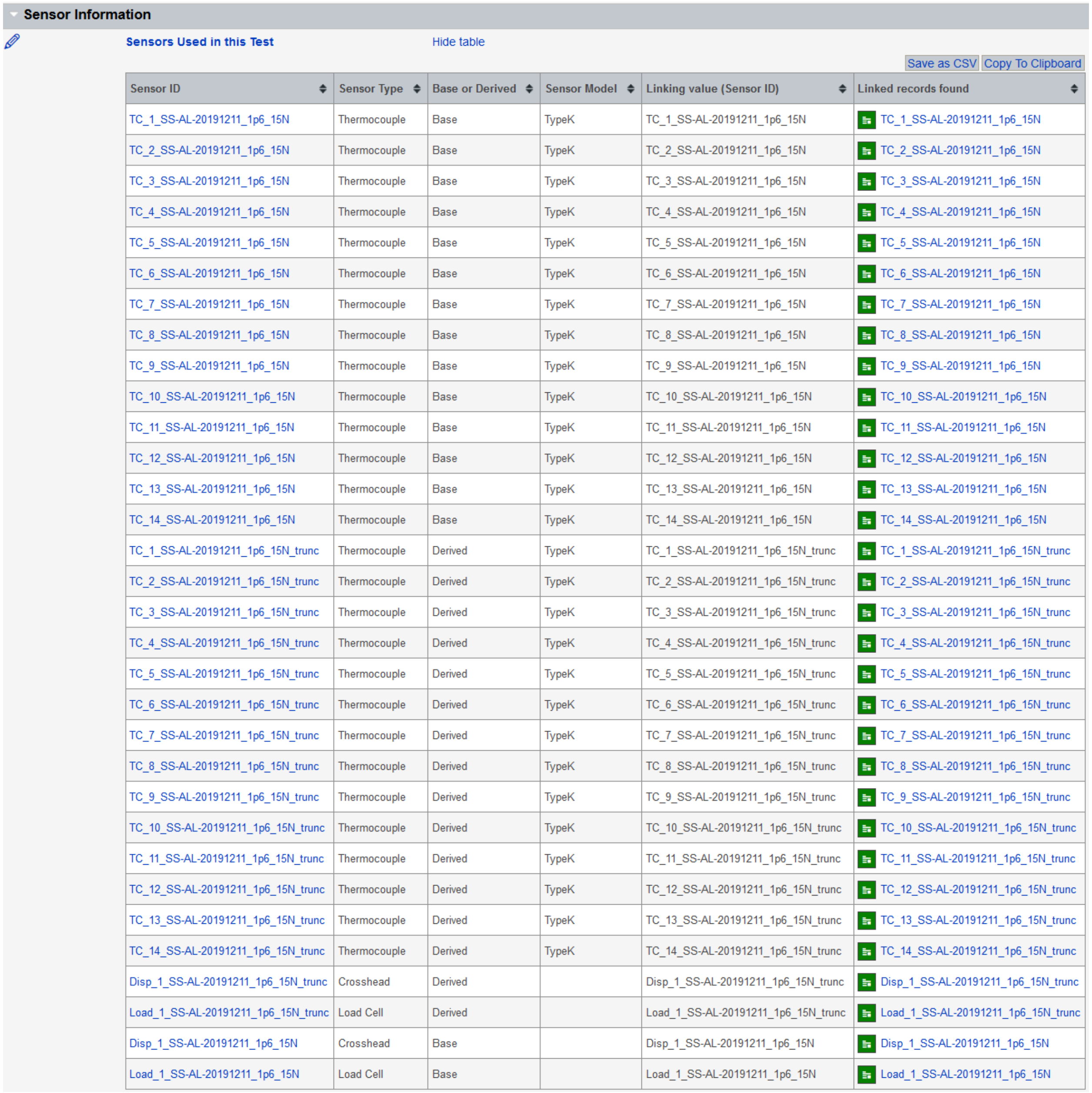
2.3.4 Tests

Test records are located one level below the test series records. Each test record must be linked with the test series record via *Test ID* attribute (see the previous section description how to create tabular links). Also, each test record must be linked to its children sensor record(s). Grouping sensor records into a test is much more intuitive than grouping various tests into a test series. In a single test record an engineer should include all sensor measurements that were done for a particular "physical" test. All sensor records linked to the same test record should share the same time attribute or there should be a relationship between time attributes for different sensors. In conclusion, the sensors linked to the same test record are not just some arbitrarily grouped sensors, but they are the results of various measurements done during the single test on the same test article(s).

In our example, each test record groups the individual sensor measurements done during the test. It is seen in **Figure 9** that during each test 14 thermocouples are used to measure the temperature at different locations on metal cylinders and on hot and cold platens. In addition, there is one displacement sensor and one force sensor. In total, there are 16 physical sensors used in each test.

As an example, tabular attribute *Sensor Used in this Test* is shown for test *SS-AL-20191211_1p6_15N* in **Figure 18**. It is seen in this figure that the total number of sensors for the test is 32 not 16. The reason the number of sensors was doubled is that the so-called derived (virtual) sensors were added to this test record. The data in the derived sensors is based on the data from the corresponding physical (base) sensors, but the data was for example truncated to save time and memory during downloading the data from database. It is allowed to truncate, smooth, or to remove outlier data points in the derived sensors. It is the project engineer's responsibility to decide if derived sensors are needed for any particular test.

Sensor records can be linked to the tabular attribute *Sensors Used in this Test* in the same way as described in the previous section. The only difference is that in this case the linking attribute is *Sensor ID* attribute in the sensor records.



Sensor Information

Sensors Used in this Test

Hide table

Sensor ID	Sensor Type	Base or Derived	Sensor Model	Linking value (Sensor ID)	Linked records found
TC_1_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_1_SS-AL-20191211_1p6_15N	TC_1_SS-AL-20191211_1p6_15N
TC_2_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_2_SS-AL-20191211_1p6_15N	TC_2_SS-AL-20191211_1p6_15N
TC_3_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_3_SS-AL-20191211_1p6_15N	TC_3_SS-AL-20191211_1p6_15N
TC_4_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_4_SS-AL-20191211_1p6_15N	TC_4_SS-AL-20191211_1p6_15N
TC_5_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_5_SS-AL-20191211_1p6_15N	TC_5_SS-AL-20191211_1p6_15N
TC_6_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_6_SS-AL-20191211_1p6_15N	TC_6_SS-AL-20191211_1p6_15N
TC_7_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_7_SS-AL-20191211_1p6_15N	TC_7_SS-AL-20191211_1p6_15N
TC_8_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_8_SS-AL-20191211_1p6_15N	TC_8_SS-AL-20191211_1p6_15N
TC_9_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_9_SS-AL-20191211_1p6_15N	TC_9_SS-AL-20191211_1p6_15N
TC_10_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_10_SS-AL-20191211_1p6_15N	TC_10_SS-AL-20191211_1p6_15N
TC_11_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_11_SS-AL-20191211_1p6_15N	TC_11_SS-AL-20191211_1p6_15N
TC_12_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_12_SS-AL-20191211_1p6_15N	TC_12_SS-AL-20191211_1p6_15N
TC_13_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_13_SS-AL-20191211_1p6_15N	TC_13_SS-AL-20191211_1p6_15N
TC_14_SS-AL-20191211_1p6_15N	Thermocouple	Base	TypeK	TC_14_SS-AL-20191211_1p6_15N	TC_14_SS-AL-20191211_1p6_15N
TC_1_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_1_SS-AL-20191211_1p6_15N_trunc	TC_1_SS-AL-20191211_1p6_15N_trunc
TC_2_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_2_SS-AL-20191211_1p6_15N_trunc	TC_2_SS-AL-20191211_1p6_15N_trunc
TC_3_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_3_SS-AL-20191211_1p6_15N_trunc	TC_3_SS-AL-20191211_1p6_15N_trunc
TC_4_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_4_SS-AL-20191211_1p6_15N_trunc	TC_4_SS-AL-20191211_1p6_15N_trunc
TC_5_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_5_SS-AL-20191211_1p6_15N_trunc	TC_5_SS-AL-20191211_1p6_15N_trunc
TC_6_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_6_SS-AL-20191211_1p6_15N_trunc	TC_6_SS-AL-20191211_1p6_15N_trunc
TC_7_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_7_SS-AL-20191211_1p6_15N_trunc	TC_7_SS-AL-20191211_1p6_15N_trunc
TC_8_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_8_SS-AL-20191211_1p6_15N_trunc	TC_8_SS-AL-20191211_1p6_15N_trunc
TC_9_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_9_SS-AL-20191211_1p6_15N_trunc	TC_9_SS-AL-20191211_1p6_15N_trunc
TC_10_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_10_SS-AL-20191211_1p6_15N_trunc	TC_10_SS-AL-20191211_1p6_15N_trunc
TC_11_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_11_SS-AL-20191211_1p6_15N_trunc	TC_11_SS-AL-20191211_1p6_15N_trunc
TC_12_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_12_SS-AL-20191211_1p6_15N_trunc	TC_12_SS-AL-20191211_1p6_15N_trunc
TC_13_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_13_SS-AL-20191211_1p6_15N_trunc	TC_13_SS-AL-20191211_1p6_15N_trunc
TC_14_SS-AL-20191211_1p6_15N_trunc	Thermocouple	Derived	TypeK	TC_14_SS-AL-20191211_1p6_15N_trunc	TC_14_SS-AL-20191211_1p6_15N_trunc
Disp_1_SS-AL-20191211_1p6_15N_trunc	Crosshead	Derived		Disp_1_SS-AL-20191211_1p6_15N_trunc	Disp_1_SS-AL-20191211_1p6_15N_trunc
Load_1_SS-AL-20191211_1p6_15N_trunc	Load Cell	Derived		Load_1_SS-AL-20191211_1p6_15N_trunc	Load_1_SS-AL-20191211_1p6_15N_trunc
Disp_1_SS-AL-20191211_1p6_15N	Crosshead	Base		Disp_1_SS-AL-20191211_1p6_15N	Disp_1_SS-AL-20191211_1p6_15N
Load_1_SS-AL-20191211_1p6_15N	Load Cell	Base		Load_1_SS-AL-20191211_1p6_15N	Load_1_SS-AL-20191211_1p6_15N

Figure 18 *Tabular attribute 'Sensors Used in this Test' in* SS-AL-20191211_1p6_15N *test record contains links to base and derived sensor records.*

As before, the individual sensor record objects can be accessed in Python using the STK interface. The script that does it is shown in **Figure 19**.

```
tbl_tests = my_db.get_table("Tests")
rec_name_tests = test_records_list[0]
#
tests_records_list = get_records(rec_name_tests, tbl_tests)
#
table_attr_name_in_tests = 'Sensors Used in this Test'
sensor_records_list = get_record_names_from_tabular(tests_records_list, \
                                table_attr_name_in_tests)
```

Figure 19 *Example script used to find the Sensor records using the tabular attribute 'Sensors Used in this Test' from the test record.*

After executing the script shown in **Figure 19** it can be verified that *sensor_records_list* variable is populated with the same record objects (sensors) as shown in **Figure 18**.

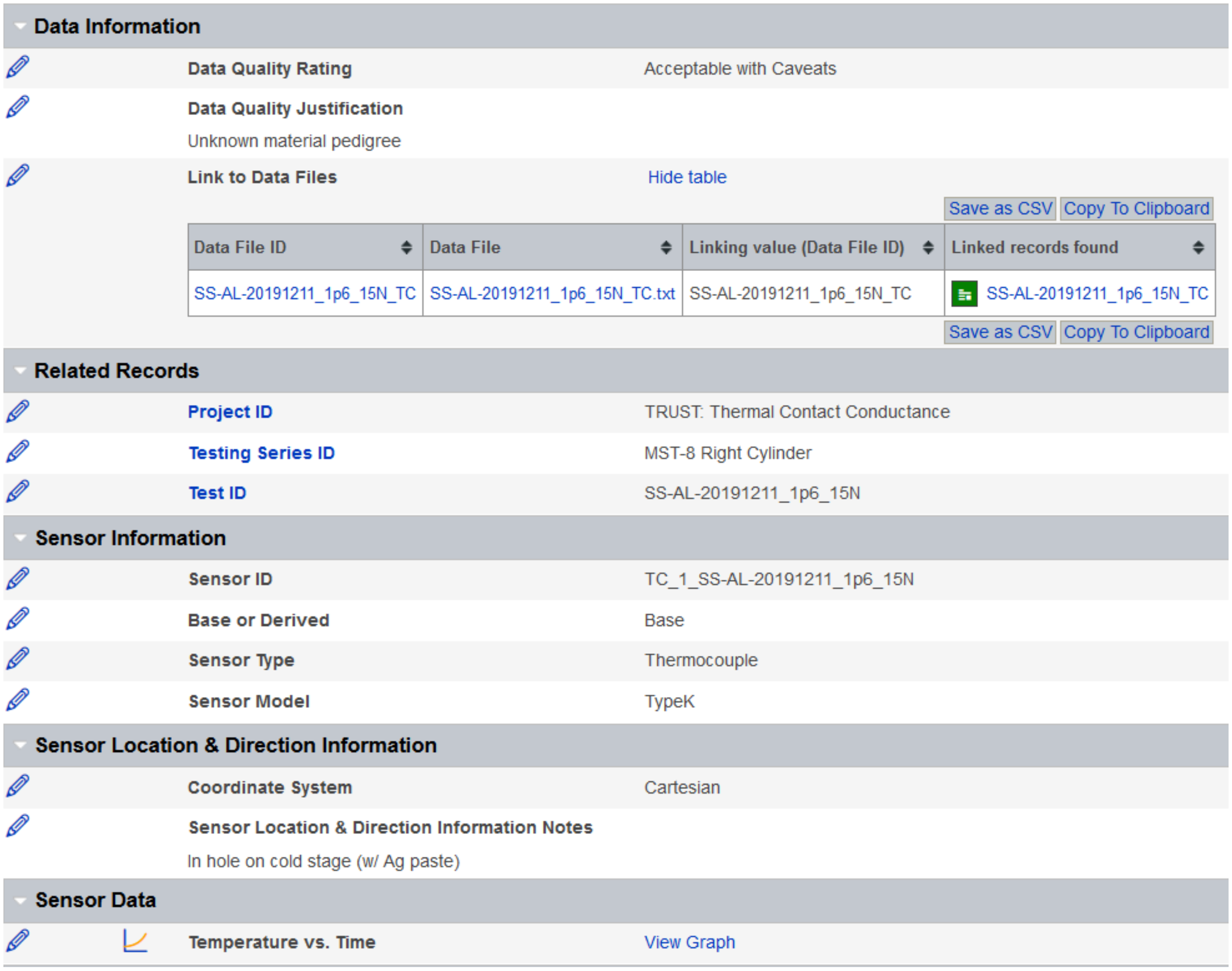
```
In [10]: sensor_records_list
Out[10]:
[['TC_1_SS-AL-20191211_1p6_15N'],
 ['TC_2_SS-AL-20191211_1p6_15N'],
 ['TC_3_SS-AL-20191211_1p6_15N'],
 ['TC_4_SS-AL-20191211_1p6_15N'],
 ['TC_5_SS-AL-20191211_1p6_15N'],
 ['TC_6_SS-AL-20191211_1p6_15N'],
 ['TC_7_SS-AL-20191211_1p6_15N'],
 ['TC_8_SS-AL-20191211_1p6_15N'],
 ['TC_9_SS-AL-20191211_1p6_15N'],
 ['TC_10_SS-AL-20191211_1p6_15N'],
 ['TC_11_SS-AL-20191211_1p6_15N'],
 ['TC_12_SS-AL-20191211_1p6_15N'],
 ['TC_13_SS-AL-20191211_1p6_15N'],
 ['TC_14_SS-AL-20191211_1p6_15N'],
 ['TC_1_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_2_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_3_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_4_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_5_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_6_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_7_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_8_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_9_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_10_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_11_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_12_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_13_SS-AL-20191211_1p6_15N_trunc'],
 ['TC_14_SS-AL-20191211_1p6_15N_trunc'],
 ['Disp_1_SS-AL-20191211_1p6_15N_trunc'],
 ['Load_1_SS-AL-20191211_1p6_15N_trunc'],
 ['Disp_1_SS-AL-20191211_1p6_15N'],
 ['Load_1_SS-AL-20191211_1p6_15N']]
```

Figure 20 *Variable sensor_records_list is populated with individual sensor record objects (same as records listed in Figure 18).*

2.3.5 Sensors

Sensor records are located one level below the test records. Sensors do not have any children records, although they may be linked to records in the Data Files table.

As an example, the *Data Information*, *Related Records*, *Sensor Information*, *Sensor Location & Direction Information* and *Sensor Data* sections of sensor TC_6_SS-AL-20191211_1p6_15N are shown in **Figure 21**. In the Data Information section, Data Quality Ranking and Data Quality Justification is provided. It is recommended that Data Quality Rating should be provided strictly to the measurements done by the current sensor only. An example of an issue that could affect an individual sensor quality rating would be an observation that the thermocouple is not fully bonded to the test article, solder connection between the thermocouple and the acquisition wire was cracking, etc. In addition to quality rating and its justification, the tabular link to the record with the Data File is also provided in the Data Information section. In this case, the Data File record contains a raw file with the results of measurements for all thermocouples bonded to the tested cylinder.



Data Information

Data Quality Rating	Acceptable with Caveats
Data Quality Justification Unknown material pedigree	
Link to Data Files	Hide table

Data File ID	Data File	Linking value (Data File ID)	Linked records found
SS-AL-20191211_1p6_15N_TC	SS-AL-20191211_1p6_15N_TC.txt	SS-AL-20191211_1p6_15N_TC	SS-AL-20191211_1p6_15N_TC

Related Records

Project ID	TRUST: Thermal Contact Conductance
Testing Series ID	MST-8 Right Cylinder
Test ID	SS-AL-20191211_1p6_15N

Sensor Information

Sensor ID	TC_1_SS-AL-20191211_1p6_15N
Base or Derived	Base
Sensor Type	Thermocouple
Sensor Model	TypeK

Sensor Location & Direction Information

Coordinate System	Cartesian
Sensor Location & Direction Information Notes In hole on cold stage (w/ Ag paste)	

Sensor Data

Temperature vs. Time	View Graph

Figure 21 *Typical information included in the sensor record window. Data for sensor TC_6_SS-AL-20191211_1p6_15N is shown.*

In the Related Records section, the IDs for parent Project, Testing Series, and Test records are provided. It is strongly recommended that the IDs for all parent records are provided in each Sensor record. Including the parent records IDs simplifies the searching of the database using STK 2.x.

As a minimum, in the Sensor Information section the project engineer should define the unique ID for the sensor (Sensor ID attribute), specify if the sensor is base or derived (Base of Derive attribute), specify sensor type (Sensor Type attribute), and provide information about sensor model (Sensor Model attribute). It is also required that information about the location and/or orientation of a sensor is provided in Sensor Location & Direction Information section.

Finally, the data measured by the sensor is saved in the Sensor Data section. In the present example, the sensor is a thermocouple that measures temperature as a function of time. Therefore, the only attribute in *Sensor Data* is the *Temperature vs. Time* functional attribute. The graph showing measured temperature vs. time is displayed as seen in **Figure 22**

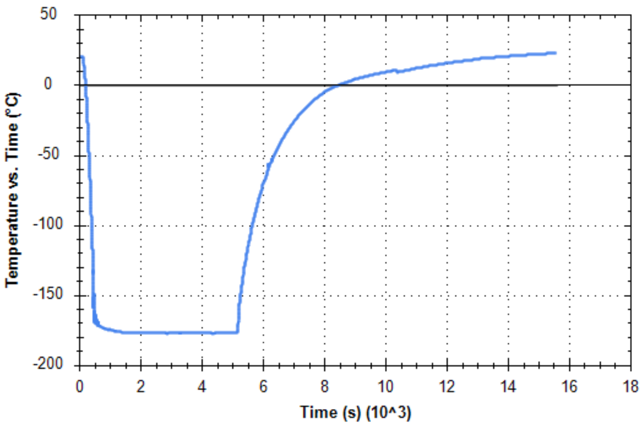


Figure 22 *Temperature vs. time plot for sensor TC_6_SS-AL-20191211_1p6_15N*

One of the main advantages of using the Python to TIMS database interface such as STK 2.x is the fact that STK 2.x can become a convenient tool to automate various tasks typically done when test data is analyzed. For example, we can easily write a script to compare measured temperature vs. time data for different thermocouples. A script that generates a composite plot for six first thermocouples in Test *SS-AL-20191211_1p6_15N* is shown in **Figure 23**.

```
tbl_sensors = my_db.get_table("Sensors")
rec_name_sensors = sensor_records_list[0:6]
#
sensors_records_list = get_records(rec_name_sensors, tbl_sensors)
#
attr_func = 'Temperature vs. Time'
funct_dict = get_functional_data_dict(sensors_records_list, attr_func)
#
#
# Create Example plots:
# --------------------
plot_file_name = 'temp_vs_time_example.png'
ax = plt.gca()
#
plt.figure(1)
for key in list(funct_dict.keys()):
    df_out = funct_dict[key]
    df_columns = list(df_out.columns)
    df_out.plot(kind='line', x=df_columns[0], y=df_columns[1], ax=ax)
plt.savefig(plot_file_name)
plt.close(1)
```

Figure 23 *Script to read and plot temperature vs. time data for the first six thermocouples from Test* SS-AL-20191211_1p6_15N

Finally, after the script shown in **Figure 23** is executed then temperature vs. time plot shown in **Figure 24** can be generated. It is not in the scope of this report to generate other examples, but as shown by the simple script in **Figure 23** Python scripting with STK 2.x can be easily extended to generate custom plots and other numerical tools for data comparison, processing, and validation.

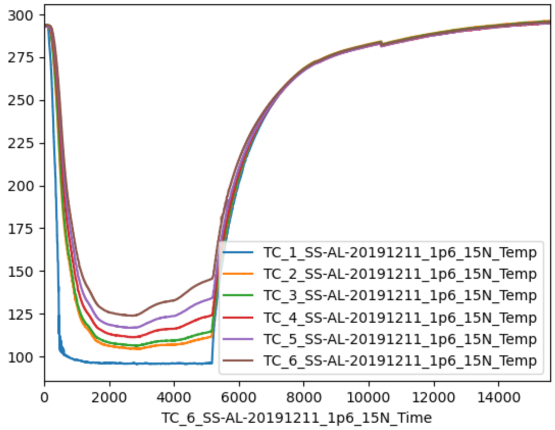


Figure 24 *Plot generated by script shown in* ***Figure 23****. Temperature vs. time curves for six sensors from Test* SS-AL-20191211_1p6_15N is shown.

3 Recommendations

The recommendations provided in this section should be followed by engineers responsible for uploading data in the TIMS database. The reviewer of the data whose name is saved in *Record Reviewed By* attribute should make sure that these recommendations were followed.

- Records belonging to the same project should be arranged such that they follow the schematic shown in Figure 12; from the bottom up:
  - Each Sensor record should have a Test record parent.
  - Each Test record should have a Test Series record parent.
  - Each Test Series record should have a Project record parent.
- Each record should have a unique identifier defined (e.g. each sensor record should have *Sensor ID* attribute defined, each test record should have *Test ID* attribute defined, etc.).
- Dynamic linking via unique IDs should always be used to link the related records.
- The names of the directories in tables should be different than any record names and any record IDs (to avoid conflict and/or confusion when the database is searched).
- Some consideration should be used to group the tests belonging to the same project under various Test Series records. How to accomplish this grouping depends on the particular tests performed for the project, but in general the situation when all tests are lumped into one *Test Series* should be avoided for all but very simple projects.
- In general, the derived sensors should be created only when specifically asked for by the analyst or in a case when the very large amount of data is saved in the base sensor and the derived sensor uses just the interpolated data.
- In general, the information in the meta-attributes in the records linked to a given project should be sufficient for other test engineer not familiar with the project to repeat the measurements (e.g. loading of a test article, placement of sensors, testing duration, data acquisition, etc.).

Additional recommendations to the TIMS database administrator:

- The value of attribute uncertainty in the Sensor record should include the maximum uncertainty of the measurement preferably in percent [%]. It is also recommended that the uncertainty attribute is changed to the short text type from the real number type. Uncertainty Units attribute is recommended to be removed from the database.
- Information for Data Processing attributes from Data Files records should be removed; these attributes were added to transfer the data to *datanode* format, but this format is no longer used.

4 Accomplishments and Future Work

In FY20 and FY21 the following milestones have been accomplished on the TIMS project as described in this report:

- Test data from various TRUST projects (e.g. Thermal Contact Conductance and Non-Linear Dynamics) was documented and saved using the TIMS database.
- Multiple staff engineers and students from W-13, MST-8, MST-7, and E-14 groups were trained to use the TIMS database and to save and document test data in TIMS.
- Dynamic linking capabilities between the records from different tables was added.
- The TIMS database was migrated to the production server.
- Python to TIMS interface (STK 2.x) was evaluated and tested; it was concluded that STK 2.x is a valuable tool for searching the database, automating the process of downloading specified attributes from TIMS to Python, and importing test and meta-data to various attributes of the TIMS database.
- Several example Python scripts that use STK 2.x package were written to illustrate how common tasks (e.g. TIMS database searching, downloading functional sensor data, etc.) could be accomplished using the Python to TIMS interface.
- The In-house developed tool Granta Query Daemon (interface between Python and TIMS) was concluded to be discontinued and STK 2.x to be used instead.
- Short explanations were written to describe how the attributes in the TIMS database should be used and what information should be saved in these attributes.

To further improve the TIMS database and to ensure that the database is routinely used by staff engineers from different testing organizations at LANL, the following tasks remain (some of these tasks are on-going):

- Release the TRUST projects records already saved in TIMS and create the new TRUST projects as new tests are being performed.
- Document and save test results other than belonging to the TRUST project tests in TIMS (e.g. recent APO-BMI verification tests performed on a part; decide where the results obtained using the Digital Image Correlation technique should be saved in TIMS).
- Add short descriptions (a.k.a. Help Files) to all attributes in the TIMS database.
- Work with the test engineers from different organizations (e.g. MST-8, E-14) to train them to use the dynamic linking between records and to use STK 2.x capabilities for saving new data results in TIMS.
- Work with the analysts in W-13 to train them to transfer the TIMS experimental data to various numerical models of parts, sub-systems, etc.
- Find an engineer in W-13 who would be responsible for reviewing and releasing new TIMS records, working with the test engineers and the analysts to make sure that the data in TIMS is complete, well documented and dynamically linked as appropriate, and easily accessible to the analysts.

5 Acknowledgements

This work was funded by the Delivery Environments Program and it is used to support the Engineering Analysis Baseline Models at Los Alamos National Laboratory.

The majority of work involved in designing and creating the TIMS database schema using Granta MI software was performed by Jillian O’Neel (MST-7).

The author would also like to thank Phil Schembri (W-13), Loren Hatler (W-13), Prabhu Khalsa (W-13), Antranik Siranosian (W-13), and Brandon Stone (W-13) for their help and contributions to the TIMS project.

6 References

Brindley, K., LeBrun, T., & Lum, S. (2020). *TRUST-EABM End-of-Year Report.* LANL Report.

Valdez, J., Cady, C., Torrez, M., Naim, T., Anghel, V., Schembri, P., & Fensin, S. (2020). *Experimental Measurement Procedure and Results for Thermal Conductivity Measurements.* LANL Report.

7 Appendix A – Python Scripts Used for Searching and Exporting TIMS Data

```
import numpy as np
import matplotlib.pyplot as plt
#
#
# Import STK Interface:
# --------------------
import GRANTA_MIScriptingToolkit as GrantaToolkit
from GRANTA_MIScriptingToolkit import granta as mpy
#
# Connect to TIMS database:
# ------------------------
#
# connect_method must be:
#   - 'crypto'
#   - 'win_password'
#
connect_method = 'crypto'                              # <--- INPUT connect_method

#your_Z_number  = str(input('Enter Your Z number: '))

your_Z_number = 'xxxxxx'
#
if connect_method=='crypto':
    import getpass
    password = getpass.getpass('Enter your crypto password: ')
    mi = mpy.connect("http://grantami/mi_servicelayer", user_name=your_Z_number, \
        password=password,  domain="lanl.gov", autologon=False)
elif connect_method=='win_password':
    import getpass
    password = getpass.getpass('Enter your WIN temporary password: ')
    #password = 'WIN_password_here'
    mi = mpy.connect("http://grantami/mi_servicelayer", user_name=your_Z_number, \
        password=password, domain="win.lanl.gov", autologon=False)
else:
    print('Incorrect value of  connect_method')
#
my_db = mi.get_db("Test Information Management System")                 # Connect to
TIMS
#
# Set Units:
# ---------
my_db.set_unit_system('SI (Consistent)', absolute_temperatures=True)
```

Figure 25 *Script used to start STK 2.x package and to connect to TIMS database*

```
def get_records(term_to_search_list, tbl):
    '''
    Input:
            - term_to_search_list:  list of names (string) of records to search in the
database
            - tbl:                  database table to be searched
    Output:
            - list of strings; each string is the name of the found record

    '''
    records_found = []
    for _ in term_to_search_list:
        if type(_)==list:
            _=_[0]
        try:
            records_aux = tbl.search_for_records_by_name(_)
            for ind in range(len(records_aux)):
                if not records_aux[ind].is_folder:
                    records_found.append(records_aux[ind])
        except IndexError:
            print('No records found')
    return records_found
#
def get_record_names_from_tabular(records_found, table_attr_name, col=0):
    '''
    Input:
            - records_found:    list of strings; each string must be the name of the
record
            - table_attr_name:  name of the attribute that points to the table to be
searched
            - col:              column number with record names
    Output:
            - list of strings; each string is the name of the found record
    '''
    names_found = []
    for _ in records_found:
        tabular_data = _.attributes[table_attr_name].data
        for aux in range(len(tabular_data)):
            names_found.append(tabular_data[aux][col][0])
            print(tabular_data[aux][col])
    return names_found
```

Figure 26 *Example script to define the custom functions get_records and get_record_names_from_tabular to search the TIMS database using the tabular links.*

```
def get_functional_data_dict(records_list, attr_func):
    '''
    Input:
            - records_list: list of .Record objects
            - attr_func:    string specifying the name of the functional data
attribute to export
    Output:
            - dictionary:   keys are names of the records from records_list
                            values are pandas dataframe objects (first column is time
and
                            second column is Force or Temperature - depending on what
fuctional
                            data attribute was specified in attr_func argument)
    -------------------------------------------------------------------------
    Warning:    This function only works with Force vs. Time and
                Temperature vs. Time functional attributes. Other attributes
                will be added.
    -------------------------------------------------------------------------
    '''
    all_data_dict = dict()
    for _ in records_list:
        if attr_func == 'Force vs. Time':
            aux_time = []
            aux_force = []
            for ind in range(len(_.attributes[attr_func].data)-1):
                aux_time.append(_.attributes[attr_func].data[ind+1][2])
                aux_force.append(_.attributes[attr_func].data[ind+1][0])
            #print(aux_time)
            col_time = _.name + '_Time'
            col_Force = _.name + '_Force'
            aux_df = pd.DataFrame({col_time:aux_time, col_Force:aux_force})
            all_data_dict.update({_.name:aux_df})
        elif attr_func == 'Temperature vs. Time':
            aux_time = []
            aux_temp = []
            for ind in range(len(_.attributes[attr_func].data)-1):
                aux_time.append(_.attributes[attr_func].data[ind+1][2])
                aux_temp.append(_.attributes[attr_func].data[ind+1][0])
            #print(aux_time)
            col_time = _.name + '_Time'
            col_temp = _.name + '_Temp'
            aux_df = pd.DataFrame({col_time:aux_time, col_temp:aux_temp})
            all_data_dict.update({_.name:aux_df})
    return all_data_dict
```

Figure 27 *Example script to define the custom function get_functional_data_dict to import sensor's functional data to Python dictionary.*

```
# PROJECT Table:
# -------------
tbl_project = my_db.get_table("Projects")
#
# Name of the project record:
# --------------------------
rec_name_project = ['TRUST: Thermal Contact Conductance']
#
# List of Project records found:
# -----------------------------
project_records_list = get_records(rec_name_project, tbl_project)
#
# Name of the tabular attribute with all linked Test Series records:
# -----------------------------------------------------------------
table_attr_name_in_project = 'Link to Testing Series Records'
#
# Find Test Series records linked from the Project record:
# -------------------------------------------------------
test_series_records_list = get_record_names_from_tabular([project_records_list[0]], \
                                table_attr_name_in_project)
# TEST SERIES Table:
# -----------------
tbl_test_series = my_db.get_table("Testing Series")
#
# Name of the Test Series record:
# ------------------------------
rec_name_test_series = test_series_records_list
#
test_series_records_list = get_records(rec_name_test_series, tbl_test_series)
#
table_attr_name_in_test_series = 'Link to Test Records'
test_records_list = get_record_names_from_tabular(test_series_records_list, \
                                table_attr_name_in_test_series)
# TESTS Table:
# -----------
tbl_tests = my_db.get_table("Tests")
rec_name_tests = test_records_list
#
tests_records_list = [get_records(rec_name_tests, tbl_tests)[0]]
#
table_attr_name_in_tests = 'Sensors Used in this Test'
sensor_records_list = get_record_names_from_tabular(tests_records_list,
table_attr_name_in_tests)
#
table_attr_name_Test_Factors = 'Test Factors'
factor_name_list = get_record_names_from_tabular(tests_records_list, \
                                table_attr_name_Test_Factors, col=0)
factor_value_list = get_record_names_from_tabular(tests_records_list, \
                                table_attr_name_Test_Factors, col=1)
#
factors_dict = {factor_name_list[i]: factor_value_list[i] for i in
range(len(factor_name_list))}
#
# SENSORS Table:
# -------------
tbl_sensors = my_db.get_table("Sensors")
rec_name_sensors = sensor_records_list[0:6]
sensors_records_list = get_records(rec_name_sensors, tbl_sensors)
#
attr_func = 'Temperature vs. Time'
funct_dict = get_functional_data_dict(sensors_records_list, attr_func)
```

Figure 28 *Script to search: Project, Test Series, Tests, and Sensor records using the tabular links.*

8 Appendix B – List and Description of Attributes

8.1 General Record Information

Section	Attribute Name	Notes
Record Information	Data Sensitivity	This attribute contains the sensitivity of unclassified data or classification and level of classified data. Often, weapon-specific data should default to at least ECI unless otherwise approved. For classified data, a DC should be listed in the attribute "Reviewed by DC".
	Record Create Date	This attribute contains the date that this record was originally created
	Record Last Modified Date	This attribute contains the date that this record was most recently modified.
	Data Originator	This is the person who supplied the data to the modifier. In some cases this is also the point of contact or principal investigator. The purpose of knowing the data originator is that sometimes the originator is an additional point of contact who may have related information.
	Point(s) of Contact	This attribute contains the main point(s) of contact for the data listed in the record. In some cases this is also the data originator or principal investigator. This is the person or people who should be contacted first if there are questions related to the information in the record. These names should be updated when people leave the laboratory.
	Record Review Date	This attribute contains the date of the most recent review of the record for technical correctness.
	Record Reviewed By	This attribute contains the name of the person who most recently reviewed the record for technical correctness.
	Name of Modifier	This attribute contains the name the person who most recently entered or uploaded data to this record. Version history can be used to track previous modifiers.
	Reviewed by DC (Derivative Classifier)	This attribute contains the name of the person who reviewed the record for classification.

Table 1. General Record Information attributes used in any record.

8.2 Project Table

Section	Attribute Name	Notes
Project Information	Project ID	This attribute contains a unique identifier for the project. This ID is used for tabular linking in other tables.
	Project Code	This attribute contains the charge code(s) used for the project.
	Funding Program Name	This attribute contains the name of the program that funded the project.
	Project Objective	This attribute contains a description of purpose/objective of the project.
	Project Category	This attribute contains the project category (ex: Acceptance, Joint Test, Local Test, etc.).
	Weapon System	This attribute contains the weapon system(s) studied in this project.
	Date Started	This attribute contains the project start date.
	Date Finished	This attribute contains the project completion date.
	Link to Project Webpage	This attribute contains a hyperlink to the project webpage (ex: Confluence, SharePoint).
	Project Information Notes	This attribute contains additional notes about the project. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.
Testing Series Records for this Project	Link to Testing Series Records	This attribute is a tabular link to the Testing Series records associated with this project. The linking value is the "Testing Series ID" attribute for the Testing Series records of interest.
	Testing Series Records for this Project Notes	This attribute contains additional notes about the testing series. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.
Parts in this Project	Link to Parts	This attribute is a tabular link to the Parts records associated with this project. The linking value is the "Part ID" attribute for the Parts records of interest.
	Parts in this Project Notes	This attribute contains additional notes about the parts. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.
Documents in this Project	Link to Documents	This attribute is a tabular link to the Documents records associated with this project. The linking value is the "Document ID" attribute for the Documents records of interest.
	Documents in this Project Notes	This attribute contains additional notes about the documents. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.

Table 2. Attributes used in Project records

8.3 Testing Series Table

Section	Attribute Name	Notes
Testing Series Information	Project ID	This attribute is the unique ID corresponding to the project of which the testing series is a part. It can also be used to create links between this and the project record.
	Testing Series ID	This attribute contains a unique identifier for the Testing Series. This ID is used for tabular linking in other tables.
	Testing Organization	This attribute contains the testing organization responsible for the testing within this series.
	Principal Investigators	This attribute contains the principal investigators for all of the testing within this series. In some cases this may include the data originator or point of contact.
	Date Started	This attribute contains the start date for testing within this series.
	Date Finished	This attribute contains the completion date for testing within this series.
	Status	This attribute contains the current status of the tests within this series (ex: Completed or In Progress).
	Parameters Varied	This attribute contains the variables or parameters that were varied throughout the testing series (ex: locations within part, temperature profile, etc).
	Related Documents & Images	This attribute is a tabular link to the files associated with this testing series. It links to the Documents table using the "Document ID" of the file of interest.
	Testing Series Information Notes	This attribute contains additional notes about the testing series. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.
Testing Series Records for this Project	Link to Test Records	This attribute is a tabular link to the Tests records associated with this project. The linking value is the "Test ID" attribute for the Tests records of interest.
	Test Matrix Information Notes	This attribute contains additional notes about the testing series. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.
Part Instances Studied	Link to Part Instances	This attribute is a tabular link to the Part Instances records associated with this project. The linking value is the "Part Instance ID" attribute for the Part Instances records of interest.
	Part Instances Studied Notes	This attribute contains additional notes about the part instances. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.

Table 3 Attributes used in Testing Series records

8.4 Tests Table

Section	Attribute Name	Notes
Related Records	Project ID	This attribute is the unique ID corresponding to the project of which the test is a part. It can also be used to create links between this and the project record.
	Testing Series ID	This attribute is the unique ID corresponding to the testing series of which the test is a part. It can also be used to create links between this and the testing series record.
Test Information	Test ID	This attribute contains a unique identifier for the Test. This ID is used for tabular linking in other tables.
	Test Date	This attribute contains the date on which the test was performed.
	Operator	This attribute contains the name or ID of the operator that performed this test.
	Room Temperature	This attribute contains the temperature of the room while the test was being performed.
	Room Humidity	This attribute contains the humidity of the room while the test was being performed.
	Test Factors	This attribute contains the Test Factors associated with this test. These are often the parameters that were varied throughout the testing series. The first column should list the factor name(s) and the second column should list the factor value(s) that were used in this particular test. For example, if the testing series involved testing different parts at different temperatures, then the factor names in the left column may be "Part Number" and "Test Temperature" and in the right column, each individual test record would contain the specific part number and temperature for that test.
	Test Temperature Table	This attribute is a local tabular attribute that can be used to capture how the temperature was adjusted during testing.
	Test Software ID	This attribute contains the ID associated with the test software. It is common for a particular test program to be given a traceable, short text ID by the testing group.
	Work Order Number	This attribute contains the work order number for the testing. It is common for testing organizations to assign a traceable work order number to a particular testing request.
	Documents	This attribute is a tabular link to the Documents records associated with this test (ex: test plan, fixture drawings, etc.). The linking value is the "Document ID" attribute for the Documents records of interest.
	Test Information Notes	This attribute contains additional notes about the test. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.
Test Results	Summary of Test Results	This attribute is a local tabular attribute that can be used to summarize the results of the test. This is usually the result that the testing series is trying to compare as the parameters are varied. The first column should list the name of the desired result (ex: "load at pad thickness 10mm" or "visual inspection of part after testing") and the second column should list the result for this particular test (ex: "2500 N" or "pass").
	Coordinate System Images	This attribute contains images that are necessary for interpretation of the attributes in the Sensor Location & Direction Information section of the sensors records. Note that there is a similar attribute in the sensors record, and that one should be used if a different image is required for each sensor in a test. If a single image for all sensors in a test is sufficient, then this attribute (in the Tests record) should be used.
	Test Results Notes	This attribute contains additional notes about the test results. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.
Sensor Information	Sensors Used in this Test	This attribute is a tabular link to the Sensors records used in this test. The linking value is the "Sensor ID" attribute for the Sensors records of interest.
	Sensor Information Notes	This attribute contains additional notes about the sensors. If there is information that is consistently being placed in this attribute, please contact the database administrator in order to determine if a new attribute should be created.

Table 4 Attributes used in Tests records

8.5 Sensors Table

Section	Attribute Name	Notes
Data Information	Data Quality Rating	This is a somewhat subjective rating of the quality of the data. Quality may be based on several factors, including: documentation of the source of material, documentation of test and regression methods, comparisons of related results to published data, ability to account for and explain anomalous data or outliers. This attribute is used to provide information of data quality for the individual sensor only, not overall quality of data collected in the entire test (e.g. data collected by other sensors).
	Data Quality Justification	This is an explanation of how the Data Quality Rating was arrived at.
	Link to Data Files	This attribute is a tabular link to the Data File records that contain the original data file corresponding to the functional data saved in this record. The linking value is the "Data File ID" attribute for the Data File records of interest.
	Data Information Notes	This attribute contains additional notes about the original data that is deemed by the project engineer to be relevant to be saved in this record.
Related Records	Project ID	This attribute is the unique ID corresponding to the project of which the test is a part. It can also be used to create links between this and the project record.
	Testing Series ID	This attribute is the unique ID corresponding to the testing series of which the test is a part. It can also be used to create links between this and the testing series record.
	Test ID	This attribute contains a unique identifier for the Test. This ID is used for tabular linking in other tables.
Sensor Information	Sensor ID	This attribute contains a unique identifier for the Sensor. This ID is used for tabular linking in other tables.
	Base or Derived	This attribute is used to document if the data saved in the sensor record is directly measured (base) or the saved data is numerically processed using other sensors (derived)
	Other Sensors Related to this Sensor	This attribute is a tabular link to the Sensors records that were used to compute the data saved in this record. The linking value is the "Sensor ID" attribute for the Sensors records of interest.
	Sensor Type	This attribute is used to document what type of sensor was used (e.g. crosshead, extensometer, load cell, thermocouple, etc.). This attribute is only applicable for base sensors.
	Sensor Model	This attribute contains information about the model name of the sensor.
	Sensor Serial Number	This attribute contains information about the serial number of the sensor.
	Alternate Sensor Names	This attribute contains information about the any alternative names of the sensor that may also be used.
	Uncertainty	This attribute contains a single value representing the uncertainty of the sensor. It is recommended that the larges applicable uncertainty value is used for measurements where the uncertainty varies depending on the measured quantity.
	Uncertainty Units	This attribute defines a unit for the uncertainty value. Note that this is a short text attribute, so if you change the database units for the database, this attribute will not be affected. Percent [%] units acceptable.
	Sensor Information Notes	This attribute contains any additional information about the sensor or how it was used in this particular test.

Table 5 Attributes used in Sensors records (1)

Section	Attribute Name	Notes
Sensor Location and Direction Information	Location Name	This attribute contains the unique location name for the sensor. The same location name for this sensor should be used in any documentation and other related attributes, if applicable.
	Coordinate System	This attribute defines the coordinate system (e.g. Cartesian, polar, etc.) used to determine the location and/or orientation (if applicable) of the sensor with respect to the test article.
	Coordinate System Images	This attribute contains images that are necessary for interpretation of the attributes in the Sensor Location & Direction Information section of the sensors record. Note that there is a similar attribute in the Tests record, and that one should be used if a single image is sufficient for locating/orienting all sensors in a test. If a separate image for each sensors in a test is necessary, then this attribute (in the Sensors record) should be used.
	Coordinate System Notes	This attribute contains any additional information about the sensor's coordinate system.
	X Location (Cartesian)	This attribute contains the coordinates of the location and/or orientation of the sensor in the coordinate system defined in Coordinate System attribute.
	Y Location (Cartesian)	
	Z Location (Cartesian)	
	Radial Distance, ρ (Cylindrical)	
	Azimuth Angle, φ (Cylindrical)	
	Height, z (Cylindrical)	
	Radius, r (Spherical)	
	Elevation Angle, θ (Spherical)	
	Azimuth Angle, φ (Spherical)	
	Sensor Direction - X	
	Sensor Direction - Y	
	Sensor Direction - Z	
	Sensor Location & Direction Information Notes	This attribute contains any additional information about the sensor's coordinate system (e.g. any assumptions, simplifications, etc.)
Calibration Information	Calibration Date	This attribute contains the date when the sensor was calibrated (if applicable). This is not the calibration date of the data acquisition system, or other "global" instruments.
	Calibration Information Notes	This attribute contains any additional information about the sensor's calibration.
Sensor Data	Force vs. Time	This attribute contains functional data collected by the individual sensor (e.g. displacement cs. time, force vs. time, temperature vs. time, etc.). Only one of these attributes may exist in a single sensor record.
	Length vs. Time	
	Temperature vs. Time	
	Column Heading	This is a heritage attribute that will be deleted.
	Sensor Data Notes	This attribute contains any additional information about the sensor's collected data (e.g. information about how the data in a derived sensor was computed).
	Summary Picture of Sensor Data	This attribute contains any image(s) that can be helpful to interpret the data collected for this sensor (e.g. graph(s) showing any trends, etc.).

Table 6 Attributes used in Sensors records (2)

8.6 Data Files Table

Section	Attribute Name	Notes
Data Information	Data Quality Rating	This is a somewhat subjective rating of the quality of the data. Quality may be based on several factors, including: documentation of the source of material, documentation of test and regression methods, comparisons of related results to published data, ability to account for and explain anomalous data or outliers. This attribute is used to provide information of data quality for the entire test (combination of data collected by individual sensors).
	Data Quality Justification	This is an explanation of how the Data Quality Rating was arrived at.
	Data File	This attribute contains the embedded raw file with the measured data.
	Data File ID	This attribute contains a unique identifier for the Data File ID. This ID is used for tabular linking in other tables.
	Link to Sensor	This attribute is a tabular link to the Sensors records that use the raw data provided in this record. Both links to base and derived sensor records are allowed. The linking value is the "Sensor ID".
	Data Information Notes	This attribute contains any additional information about the raw data file saved in this record.
Information for Data Processing	File Start Time Stamp	These attributes contain heritage information to export data to datanode format. The datanode format is planned to be no longer used in the future.
	Header	
	Delimiter	
	Time Index Column Name	
	Segment Number	
	Segment Description	
	Time Index Format	
	Sensor Information Notes	

Table 7 Attributes used in Data Files records

8.7 Parts Table

Section	Attribute Name	Notes
Part Information	Weapon System	This attribute contains information about the Weapon System this Part belongs to.
	Part Name	This attribute contains the name of the Part.
	Part ID	This attribute contains a unique identifier for the Part. This ID is used for tabular linking in other tables.
	Description	This attribute contains the brief description of the Part drawing.
	Drawing Number	This attribute contains the number of the drawing as it appears on the drawing.
	Drawing File	This attribute contains the embedded drawing file.
	Part Information Notes	This attribute contains any notes the modifier wishes the user to see about this Part.
	Link to PDMLink	This attribute is a link to the PDMLink object with this Part. The linking value is the URL link to PDMLink object.
	Link to Part Instances	This attribute is a tabular link to the Part Instance records. The linking value is the "Part Instance ID" attribute for the Sensors records of interest.

Table 8 Attributes used in Parts records

8.8 Part Instances Table

Section	Attribute Name	Notes
Part Information	Serial Number	This attribute contains a serial number of the Part Instance.
	Part Instance ID	This attribute contains a unique identifier for the Part Instance. This ID is used for tabular linking in other tables.
	Part Information Notes	This attribute contains any notes the modifier wishes the user to see about the Part Instance.

Table 9 Attributes used in Part Instances records

8.9 Documents Table

Section	Attribute Name	Notes
Document Information	Document ID	This attribute contains a unique identifier for the Document. This ID is used for tabular linking in other tables.
	Document Name	This attribute contains the name/title of the document as it appears on the document.
	Document Number	This attribute contains the document number of the document as it appears on the document.
	Author(s)	This attribute contains the name of the author or a list of author names.
	Document Date	This attribute contains the date that the document was created
	Document (embedded)	This attribute was used in legacy records but is not used for new records.
	Document Information Notes	This attribute contains any notes the modifier wishes the user to see about this document.

Table 10 Attributes used in Documents records

9 Appendix C – List of Acronyms and Abbreviations

The following acronyms and abbreviations were used in this document.

EABM	Engineering Analysis and Baseline Models
ECI	Export Controlled Information
E-14	Test Engineering Group
FE	Finite Element
GQD	Granta Query Daemon
MIMS	Material Information Management System
MST	Materials Science and Technology Division
MST-7	Engineering Materials Group
MST-8	Materials Science in Radiation and Dynamic Extremes Group
OUO	Official Use Only
SAE-3	Software and Applications Engineering Division
STK	Scripting Tool-Kit
TIMS	Test Information Management System
VS	Validation Suite
W-13	Advanced Engineering and Analysis Group